深入浅出iPhone开发（影印版）

Covers iPhone 3.1 SDK

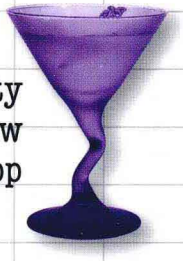
Head First iPhone Development

Master Objective-C, Interface Builder, and XCode

Design top-selling apps

Tap into the iPhone's GPS and camera

See how Mike saved his love life with an iPhone Twitter app



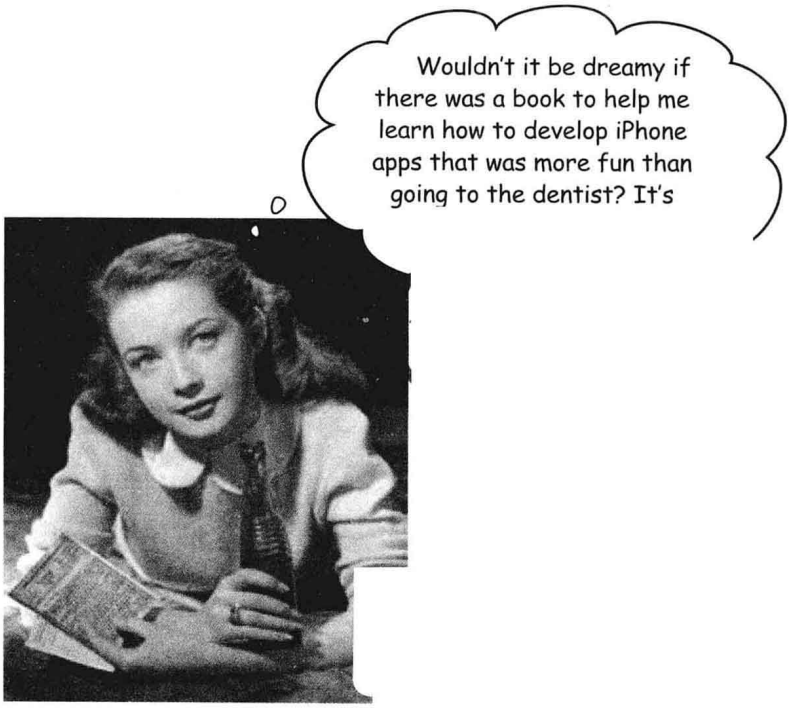
Mix up a tasty multi-view bartending app

O'REILLY® 東南大學出版社

Dan Pilone & Tracey Pilone 著

深入浅出iPhone开发(影印版)

Head First iPhone Development



Dan Pilone
Tracey Pilone

O'REILLY®

Beijing • Cambridge • Köln • Sebastopol • Taipei • Tokyo

O'Reilly Media, Inc.授权东南大学出版社出版

东南大学出版社

图书在版编目（CIP）数据

深入浅出iPhone开发：英文/（美）皮隆（Pilone, D.），（美）皮隆（Pilone, T.）著. —影印本. —南京：东南大学出版社，2010.6

书名原文：Head First iPhone Development

ISBN 978-7-5641-2267-6

Ⅰ. ①深… Ⅱ. ①皮… ②皮… Ⅲ. ①移动通信－携带电话机－应用程序－程序设计－英文 Ⅳ. ① TN929.53

中国版本图书馆CIP数据核字（2010）第089079号

江苏省版权局著作权合同登记

图字：10-2010-160号

深入浅出iPhone开发（影印版）

出版发行：东南大学出版社
地　　址：南京四牌楼2号　　邮编：210096
出 版 人：江　汉
网　　址：http://press.seu.edu.cn
电子邮件：press@seu.edu.cn
印　　刷：扬中市印刷有限公司
开　　本：787毫米×980毫米　12开本
印　　张：46印张
字　　数：769千字
版　　次：2010年6月第1版
印　　次：2010年6月第1次印刷
书　　号：ISBN 978-7-5641-2267-6
印　　数：1~2000册
定　　价：98.00元（册）

本社图书若有印装质量问题，请直接与读者服务部联系。电话（传真）：025-83792328

Advance Praise for *Head First iPhone Development*

"The great thing about this book is its simple, step-by-step approach. It doesn't try to teach everything—it just launches you right into building iPhone applications in a friendly, conversational way. It's a fantastic book for people who already know how to write code and just want to get straight into the meat of building iPhone applications."

— Eric Shephard, owner of Syndicomm

"*Head First iPhone Development* was clearly crafted to get you easily creating, using and learning iPhone technologies without needing a lot of background with Macintosh development tools."

— Joe Heck, Seattle Xcoders founder

"This book is infuriating! Some of us had to suffer and learn iPhone development 'the hard way,' and we're bitter that the jig is up."

— Mike Morrison, Stalefish Labs founder

"*Head First iPhone Development* continues the growing tradition of taking complex technical subjects and increasing their accessibility without reducing the depth and scope of the content. iPhone Development is a steep learning curve to climb by any measure, but with *Head First iPhone Development*, that curve is accompanied with pre-rigged ropes, a harness, and an experienced guide! I recommend this book for anyone who needs to rapidly improve their understanding of developing for this challenging and exciting platform."

— Chris Pelsor, snogboggin.com

Praise for other *Head First* books

"*Head First Object Oriented Analysis and Design* is a refreshing look at subject of OOAD. What sets this book apart is its focus on learning. The authors have made the content of OOAD accessible, usable for the practitioner."

— Ivar Jacobson, Ivar Jacobson Consulting

"I just finished reading HF OOA&D and I loved it! The thing I liked most about this book was its focus on why we do OOA&D-to write great software!"

— Kyle Brown, Distinguished Engineer, IBM

"Hidden behind the funny pictures and crazy fonts is a serious, intelligent, extremely well-crafted presentation of OO Analysis and Design. As I read the book, I felt like I was looking over the shoulder of an expert designer who was explaining to me what issues were important at each step, and why."

— Edward Sciore, Associate Professor, Computer Science Department, Boston College

"All in all, *Head First Software Development* is a great resource for anyone wanting to formalise their programming skills in a way that constantly engages the reader on many different levels."

— Andy Hudson, Linux Format

"If you're a new software developer, *Head First Software Development* will get you started off on the right foot. And if you're an experienced (read: long-time) developer, don't be so quick to dismiss this..."

— Thomas Duff, Duffbert's Random Musings

"There's something in *Head First Java* for everyone. Visual learners, kinesthetic learners, everyone can learn from this book. Visual aids make things easier to remember, and the book is written in a very accessible style—very different from most Java manuals...*Head First Java* is a valuable book. I can see the *Head First* books used in the classroom, whether in high schools or adult ed classes. And I will definitely be referring back to this book, and referring others to it as well."

— Warren Kelly, Blogcritics.org, March 2006

"Another nice thing about *Head First Java, 2nd Edition* is that it whets the appetite for more. With later coverage of more advanced topics such as Swing and RMI, you just can't wait to dive into those APIs and code that flawless, 100000-line program on java.net that will bring you fame and venture-capital fortune. There's also a great deal of material, and even some best practices, on networking and threads—my own weak spot. In this case, I couldn't help but crack up a little when the authors use a 1950s telephone operator—yeah, you got it, that lady with a beehive hairdo that manually hooks in patch lines—as an analogy for TCP/IP ports... you really should go to the bookstore and thumb through *Head First Java, 2nd Edition*. Even if you already know Java, you may pick up a thing or two. And if not, just thumbing through the pages is a great deal of fun."

— Robert Eckstein, Java.sun.com, April 2005

"Of course it's not the range of material that makes *Head First Java* stand out, it's the style and approach. This book is about as far removed from a computer science textbook or technical manual as you can get. The use of cartoons, quizzes, fridge magnets (yep, fridge magnets ...). And, in place of the usual kind of reader exercises, you are asked to pretend to be the compiler and compile the code, or perhaps to piece some code together by filling in the blanks or ... you get the picture... The first edition of this book was one of our recommended titles for those new to Java and objects. This new edition doesn't disappoint and rightfully steps into the shoes of its predecessor. If you are one of those people who falls asleep with a traditional computer book then this one is likely to keep you awake and learning."

— TechBookReport.com, June 2005

"*Head First Web Design* is your ticket to mastering all of these complex topics, and understanding what's really going on in the world of web design...If you have not been baptized by fire in using something as involved as Dreamweaver, then this book will be a great way to learn good web design. "

— Robert Pritchett, MacCompanion, April 2009 Issue

"Is it possible to learn real web design from a book format? *Head First Web Design* is the key to designing user-friendly sites, from customer requirements to hand-drawn storyboards to online sites that work well. What sets this apart from other 'how to build a web site' books is that it uses the latest research in cognitive science and learning to provide a visual learning experience rich in images and designed for how the brain works and learns best. The result is a powerful tribute to web design basics that any general-interest computer library will find an important key to success."

— Diane C. Donovan, California Bookwatch: The Computer Shelf

"I definitely recommend *Head First Web Design* to all of my fellow programmers who want to get a grip on the more artistic side of the business. "

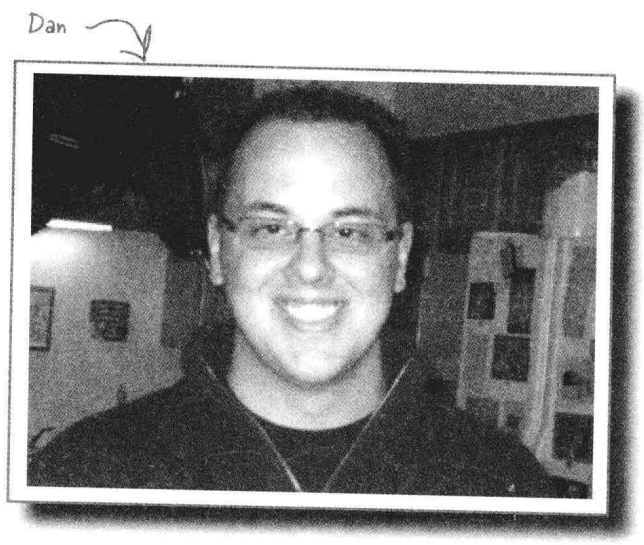
— Claron Twitchell, UJUG

To Dan, my best friend and husband, and Vinny and Nick, the best boys a mother could ask for.

—Tracey

This book is dedicated to my family: my parents who made all of this possible, my brothers who keep challenging me, and my wife and sons, who don't just put up with it—they help make it happen.

—Dan



Dan Pilone is a Software Architect for Vangent, Inc., and has led software development teams for the Naval Research Laboratory, UPS, Hughes, and NASA. He's taught graduate and undergraduate Software Engineering at Catholic University in Washington, D.C.

Dan's previous Head First books are *Head First Software Development* and *Head First Algebra*, so he's used to them being a little out of the ordinary, but this is the first book to involve bounty hunters. Even scarier was watching Tracey shift to become a night owl and Apple fan-girl to get this book done.

Dan's degree is in Computer Science with a minor in Mathematics from Virginia Tech and he is one of the instructors for the O'Reilly iPhone Development Workshop.

Tracey Pilone would first like to thank her co-author and husband for sharing another book and being relentless in his willingness to stay up late to get things right.

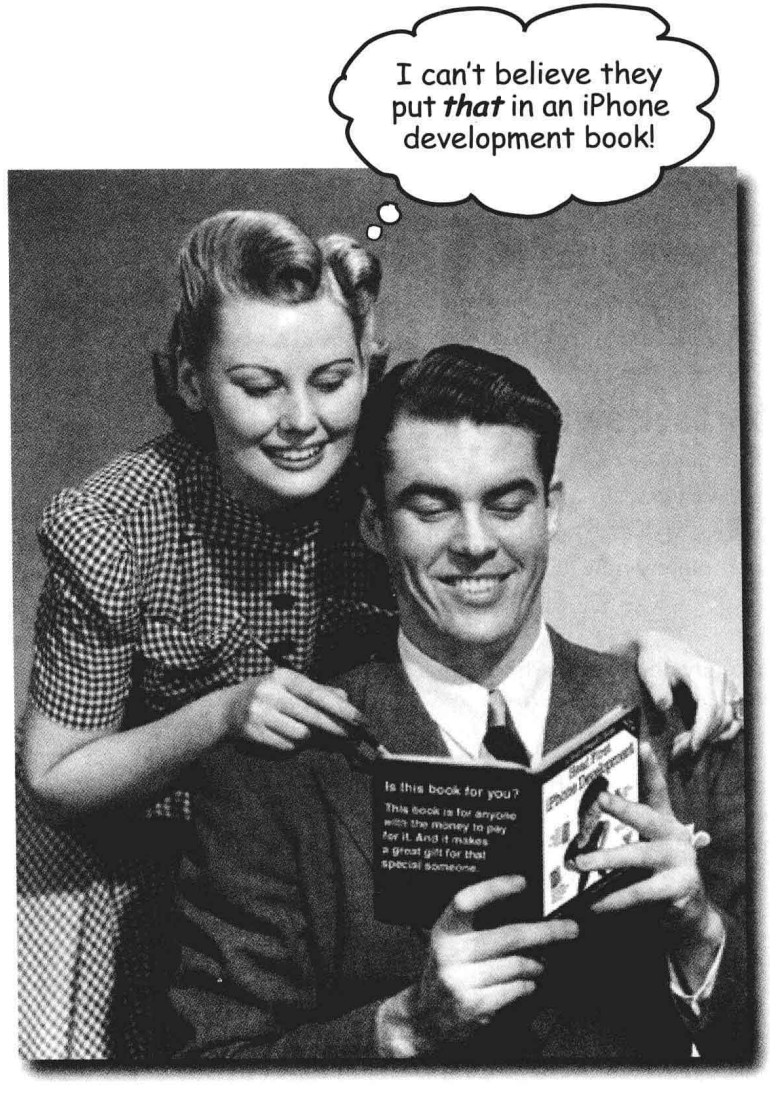
She is a freelance technical writer who supports mission planning and RF analysis software for the Navy, and is the author of *Head First Algebra*.

Before becoming a writer, she spent several years working as a construction manager on large commercial construction sites around Washington, D.C. There she was part of a team responsible for coordinating the design and construction of office buildings, using engineering and management skills that somehow all came in handy writing Head First books.

She has a Civil Engineering degree from Virginia Tech, holds a Professional Engineer's License, and received a Masters of Education from the University of Virginia.

how to use this book

Intro



In this section, we answer the burning question:
"So why DID they put that in an iPhone development book?"

Who is this book for?

If you can answer "yes" to all of these:

1. Do you have previous development experience?
2. Do you want to **learn**, **understand**, **remember**, and ***apply*** important iPhone design and development concepts so that you can write your own iPhone apps, and start selling them in the App Store?
3. Do you prefer **stimulating dinner party conversation** to **dry**, **dull**, **academic lectures**?

It definitely helps if you've already got some object-oriented chops, too. Experience with Mac development is helpful, but definitely not required.

this book is for you.

Who should probably back away from this book?

If you can answer "yes" to any of these:

1. Are you **completely new** to software development?
2. Are you already developing iPhone apps and looking for a ***reference*** **book** on Objective-C?
3. Are you **afraid to try something different**? Would you rather have a root canal than mix stripes with plaid? Do you believe that a technical book can't be serious if there's a bounty hunter in it?

Check out Head First Java for an excellent introduction to object-oriented development, and then come back and join us in iPhoneville.

this book is not for you.

[Note from marketing: this book is for anyone with a credit card. Or cash. Cash is nice, too – Ed]

We know what you're thinking.

"How can *this* be a serious iPhone development book?"

"What's with all the graphics?"

"Can I actually *learn* it this way?"

And we know what your *brain* is thinking.

Your brain craves novelty. It's always searching, scanning, *waiting* for something unusual. It was built that way, and it helps you stay alive.

So what does your brain do with all the routine, ordinary, normal things you encounter? Everything it *can* to stop them from interfering with the brain's *real* job—recording things that *matter*. It doesn't bother saving the boring things; they never make it past the "this is obviously not important" filter.

How does your brain *know* what's important? Suppose you're out for a day hike and a tiger jumps in front of you. What happens inside your head and body?

Neurons fire. Emotions crank up. *Chemicals surge.*

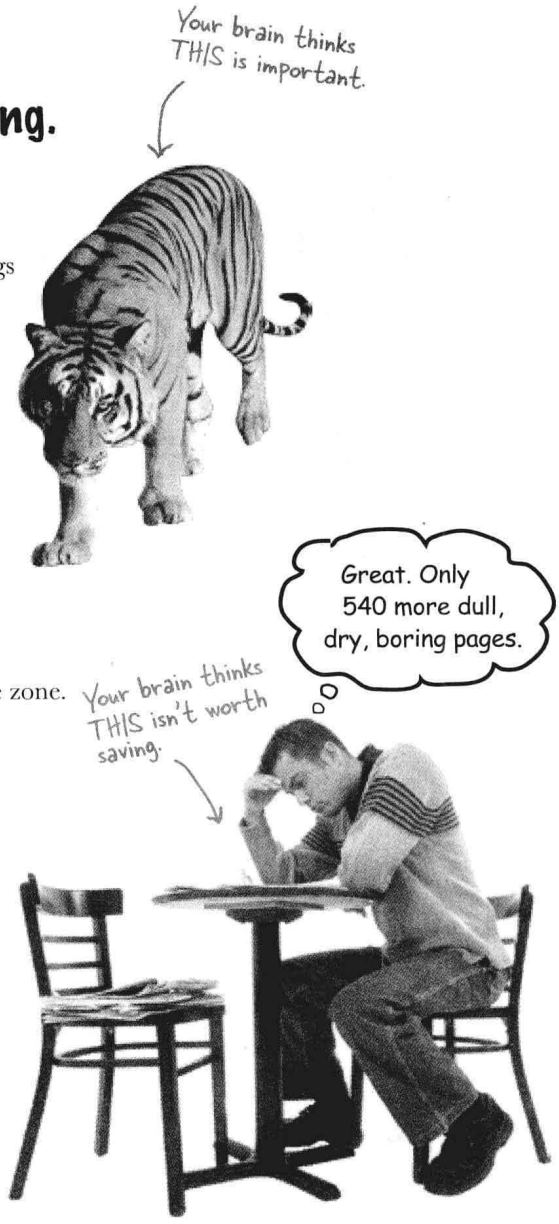
And that's how your brain knows...

This must be important! Don't forget it!

But imagine you're at home, or in a library. It's a safe, warm, tiger-free zone. You're studying. Getting ready for an exam. Or trying to learn some tough technical topic your boss thinks will take a week, ten days at the most.

Just one problem. Your brain's trying to do you a big favor. It's trying to make sure that this *obviously* non-important content doesn't clutter up scarce resources. Resources that are better spent storing the really *big* things. Like tigers. Like the danger of fire. Like how you should never again snowboard in shorts.

And there's no simple way to tell your brain, "Hey brain, thank you very much, but no matter how dull this book is, and how little I'm registering on the emotional Richter scale right now, I really *do* want you to keep this stuff around."



We think of a "Head First" reader as a learner.

So what does it take to *learn* something? First, you have to *get* it, then make sure you don't *forget* it. It's not about pushing facts into your head. Based on the latest research in cognitive science, neurobiology, and educational psychology, *learning* takes a lot more than text on a page. We know what turns your brain on.

Some of the Head First learning principles:

Make it visual. Images are far more memorable than words alone, and make learning much more effective (up to 89% improvement in recall and transfer studies). It also makes things more understandable.

Put the words within or near the graphics they relate to, rather than on the bottom or on another page, and learners will be up to *twice* as likely to solve problems related to the content.

Use a conversational and personalized style. In recent studies, students performed up to 40% better on post-learning tests if the content spoke directly to the reader, using a first-person, conversational style rather than taking a formal tone. Tell stories instead of lecturing. Use casual language. Don't take yourself too seriously. Which would *you* pay more attention to: a stimulating dinner party companion, or a lecture?

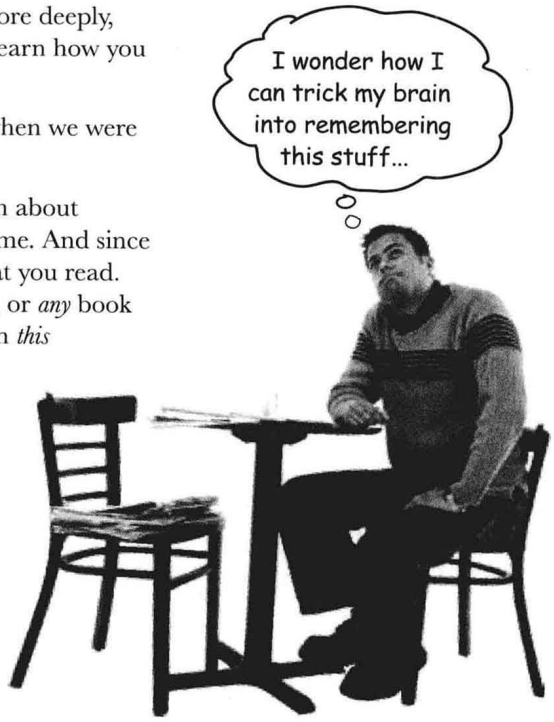
Get the learner to think more deeply. In other words, unless you actively flex your neurons, nothing much happens in your head. A reader has to be motivated, engaged, curious, and inspired to solve problems, draw conclusions, and generate new knowledge. And for that, you need challenges, exercises, and thought-provoking questions, and activities that involve both sides of the brain and multiple senses.

Get—and keep—the reader's attention. We've all had the "I really want to learn this but I can't stay awake past page one" experience. Your brain pays attention to things that are out of the ordinary, interesting, strange, eye-catching, unexpected. Learning a new, tough, technical topic doesn't have to be boring. Your brain will learn much more quickly if it's not.

Touch their emotions. We now know that your ability to remember something is largely dependent on its emotional content. You remember what you care about. You remember when you *feel* something. No, we're not talking heart-wrenching stories about a boy and his dog. We're talking emotions like surprise, curiosity, fun, "what the...?", and the feeling of "I Rule!" that comes when you solve a puzzle, learn something everybody else thinks is hard, or realize you know something that "I'm more technical than thou" Bob from engineering *doesn't*.

Metacognition: thinking about thinking

If you really want to learn, and you want to learn more quickly and more deeply, pay attention to how you pay attention. Think about how you think. Learn how you learn.



Most of us did not take courses on metacognition or learning theory when we were growing up. We were *expected* to learn, but rarely *taught* to learn.

But we assume that if you're holding this book, you really want to learn about iPhone development. And you probably don't want to spend a lot of time. And since you're going to build more apps in the future, you need to *remember* what you read. And for that, you've got to *understand* it. To get the most from this book, or *any* book or learning experience, take responsibility for your brain. Your brain on *this* content.

The trick is to get your brain to see the new material you're learning as Really Important. Crucial to your well-being. As important as a tiger. Otherwise, you're in for a constant battle, with your brain doing its best to keep the new content from sticking.

So just how *DO* you get your brain to think that iPhone development is a hungry tiger?

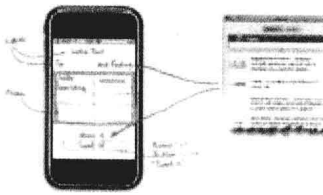
There's the slow, tedious way, or the faster, more effective way. The slow way is about sheer repetition. You obviously know that you *are* able to learn and remember even the dullest of topics if you keep pounding the same thing into your brain. With enough repetition, your brain says, "This doesn't *feel* important to him, but he keeps looking at the same thing *over* and *over* and *over*, so I suppose it must be."

The faster way is to do ***anything that increases brain activity,*** especially different *types* of brain activity. The things on the previous page are a big part of the solution, and they're all things that have been proven to help your brain work in your favor. For example, studies show that putting words *within* the pictures they describe (as opposed to somewhere else in the page, like a caption or in the body text) causes your brain to try to makes sense of how the words and picture relate, and this causes more neurons to fire. More neurons firing = more chances for your brain to *get* that this is something worth paying attention to, and possibly recording.

A conversational style helps because people tend to pay more attention when they perceive that they're in a conversation, since they're expected to follow along and hold up their end. The amazing thing is, your brain doesn't necessarily *care* that the "conversation" is between you and a book! On the other hand, if the writing style is formal and dry, your brain perceives it the same way you experience being lectured to while sitting in a roomful of passive attendees. No need to stay awake.

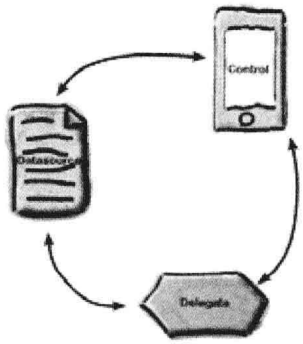
But pictures and conversational style are just the beginning.

Here's what WE did:



We used ***pictures***, because your brain is tuned for visuals, not text. As far as your brain's concerned, a picture really *is* worth a thousand words. And when text and pictures work together, we embedded the text *in* the pictures because your brain works more effectively when the text is *within* the thing the text refers to, as opposed to in a caption or buried in the text somewhere.

We used ***redundancy***, saying the same thing in *different* ways and with different media types, and *multiple senses*, to increase the chance that the content gets coded into more than one area of your brain.



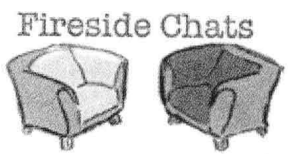
We used concepts and pictures in ***unexpected*** ways because your brain is tuned for novelty, and we used pictures and ideas with at least *some* ***emotional*** *content*, because your brain is tuned to pay attention to the biochemistry of emotions. That which causes you to *feel* something is more likely to be remembered, even if that feeling is nothing more than a little ***humor***, ***surprise***, or ***interest.***

We used a personalized, ***conversational style***, because your brain is tuned to pay more attention when it believes you're in a conversation than if it thinks you're passively listening to a presentation. Your brain does this even when you're *reading*.

We included loads of ***activities***, because your brain is tuned to learn and remember more when you ***do*** things than when you *read* about things. And we made the exercises challenging-yet-do-able, because that's what most people prefer.

We used ***multiple learning styles***, because *you* might prefer step-by-step procedures, while someone else wants to understand the big picture first, and someone else just wants to see an example. But regardless of your own learning preference, *everyone* benefits from seeing the same content represented in multiple ways.

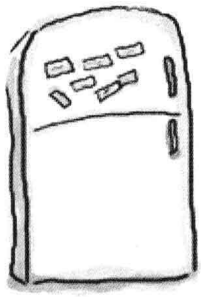
We include content for ***both sides of your brain***, because the more of your brain you engage, the more likely you are to learn and remember, and the longer you can stay focused. Since working one side of the brain often means giving the other side a chance to rest, you can be more productive at learning for a longer period of time.



And we included ***stories*** and exercises that present ***more than one point of view,*** because your brain is tuned to learn more deeply when it's forced to make evaluations and judgments.

We included ***challenges***, with exercises, and by asking ***questions*** that don't always have a straight answer, because your brain is tuned to learn and remember when it has to *work* at something. Think about it—you can't get your *body* in shape just by *watching* people at the gym. But we did our best to make sure that when you're working hard, it's on the *right* things. That ***you're not spending one extra dendrite*** processing a hard-to-understand example, or parsing difficult, jargon-laden, or overly terse text.

We used ***people***. In stories, examples, pictures, etc., because, well, because *you're* a person. And your brain pays more attention to *people* than it does to *things*.



Cut this out and stick it on your refrigerator.

Here's what YOU can do to bend your brain into submission

So, we did our part. The rest is up to you. These tips are a starting point; listen to your brain and figure out what works for you and what doesn't. Try new things.

1 Slow down. The more you understand, the less you have to memorize.

Don't just *read*. Stop and think. When the book asks you a question, don't just skip to the answer. Imagine that someone really *is* asking the question. The more deeply you force your brain to think, the better chance you have of learning and remembering.

2 Do the exercises. Write your own notes.

We put them in, but if we did them for you, that would be like having someone else do your workouts for you. And don't just *look* at the exercises. **Use a pencil.** There's plenty of evidence that physical activity *while* learning can increase the learning.

3 Read the "There are No Dumb Questions"

That means all of them. They're not optional sidebars—***they're part of the core content!*** Don't skip them.

4 Make this the last thing you read before bed. Or at least the last challenging thing.

Part of the learning (especially the transfer to long-term memory) happens *after* you put the book down. Your brain needs time on its own, to do more processing. If you put in something new during that processing time, some of what you just learned will be lost.

5 Drink water. Lots of it.

Your brain works best in a nice bath of fluid. Dehydration (which can happen before you ever feel thirsty) decreases cognitive function.

6 Talk about it. Out loud.

Speaking activates a different part of the brain. If you're trying to understand something, or increase your chance of remembering it later, say it out loud. Better still, try to explain it out loud to someone else. You'll learn more quickly, and you might uncover ideas you hadn't known were there when you were reading about it.

7 Listen to your brain.

Pay attention to whether your brain is getting overloaded. If you find yourself starting to skim the surface or forget what you just read, it's time for a break. Once you go past a certain point, you won't learn faster by trying to shove more in, and you might even hurt the process.

8 Feel something!

Your brain needs to know that this *matters*. Get involved with the stories. Make up your own captions for the photos. Groaning over a bad joke is *still* better than feeling nothing at all.

Read me

This is a learning experience, not a reference book. We deliberately stripped out everything that might get in the way of learning whatever it is we're working on at that point in the book. And the first time through, you need to begin at the beginning, because the book makes assumptions about what you've already seen and learned.

We start off by building an app in the very first chapter.

Believe it or not, even if you've never developed for the iPhone before, you can jump right in and starting building apps. You'll also learn your way around the tools used for iPhone development.

We don't worry about preparing your app to submit to the App Store until the end of book.

In this book, you can get on with the business of learning how to create iPhone apps without stressing over the packaging and distribution of your app out of the gate. But, we know that's what everyone who wants to build an iPhone app ultimately wants to do, so we cover that process (and all its glorious gotchas) in an Appendix at the end.

We focus on what you can build and test on the simulator.

The iPhone SDK comes with a great (and free!) tool for testing your apps on your computer. The simulator lets you try out your code without having to worry about getting it in the app store or on a real device. But, it also has its limits. There's some cool iPhone stuff you just can't test on the simulator, like the accelerometer and compass. So we don't cover those kinds of things in very much detail in this book since we want to make sure you're creating and testing apps quickly and easily.

The activities are NOT optional.

The exercises and activities are not add-ons; they're part of the core content of the book. Some of them are to help with memory, some are for understanding, and some will help you apply what you've learned. ***Don't skip the exercises.*** Even crossword puzzles are important—they'll help get concepts into your brain exam. But more importantly, they're good for giving your brain a chance to think about the words and terms you've been learning in a different context.

The redundancy is intentional and important.

One distinct difference in a Head First book is that we want you to *really* get it. And we want you to finish the book remembering what you've learned. Most reference books don't have retention and recall as a goal, but this book is about *learning*, so you'll see some of the same concepts come up more than once.

The Brain Power exercises don't have answers.

For some of them, there is no right answer, and for others, part of the learning experience of the Brain Power activities is for you to decide if and when your answers are right. In some of the Brain Power exercises, you will find hints to point you in the right direction.

System requirements

To develop for the iPhone, you need an Intel-based Mac, period. We wrote this book using Snow Leopard and Xcode 3.2. If you are running Leopard with an older version of Xcode, we tried to point out where there were places that would trip you up. For some of the more advanced capabilities, like the accelerometer and the camera, you'll need an actual iPhone or iPod Touch and to be a registered developer. In Chapter 1, we point you in the direction to get the SDK and Apple documentation, so don't worry about that for now.